EXPERIÊNCIA 5:

IMPLEMENTAÇÃO DE UM CRONÔMETRO COM INTERRUPÇÃO

Autores: Prof. Dr. Marco Túlio Carvalho de Andrade, Prof. Dr. Carlos Eduardo Cugnasca, Prof Dr. André Riyuiti Hirakawa , Prof. Dr. Paulo Sérgio Cugnasca

Colaborador: Prof. Dr. Jorge Kinoshita

Versão 1.3.1 - 05/2005

1. OBJETIVO

Esta experiência tem como objetivo a familiarização com os mecanismos de interrupção suportados pela família de microcontroladores 8051, utilizada na Placa Experimental.

Pretende-se que ao final desta o aluno seja capaz de habilitar interrupções, projetar rotinas de tratamento de interrupções associando-as aos endereços pertinentes, configurar periféricos para geração de interrupções e utilizar uma interface de entrada e saída programável (um módulo com display de cristal líquido).

2. TRANSFERÊNCIA DE DADOS POR INTERRUPÇÃO

Sistemas baseados em microprocessadores requerem a interligação com diversos dispositivos periféricos, tais como portas de entrada/saída paralelas, canais de comunicação seriais, controladores de teclado e display, conversores A/D, temporizadores, etc. Muitos desses periféricos requerem tratamento por parte do microprocessador em instantes de tempo periódicos, enquanto que outros são utilizados em aplicações que exigem pronto atendimento do microprocessador em instantes aleatórios, como por exemplo, sistemas e equipamentos projetados para operar em **tempo real**. Os mecanismos de atendimento normalmente permitem que se priorize o atendimento dos eventos, em situações de ocorrência simultânea de mais de uma solicitação.

Para se implementar a transferência de dados entre os microprocessadores e os dispositivos periféricos, são utilizadas diversas técnicas, escolhidas em função das particularidades de cada aplicação. Algumas das principais técnicas são: Transferência Incondicional, Tranferência Condicional ("Wait-for-Flag"), Transferência Programada/Amostragem, Interrupção e Transferência Via Acesso Direto à Memória (ADM). Estas técnicas foram descritas na Experiência 4 ("Utilização de Timer"), sendo que a técnica **Interrupção** será o alvo desta experiência.

Para se tornar mais eficiente o tratamento de periféricos que solicitam operações do microprocessador assincronamente, ou até mesmo periodicamente, existe o recurso denominado **interrupção**. Em geral, todo microprocessador apresenta sinais de interrupção e, eventualmente, é necessário algum circuito externo de controle adicional. Utilizando os sinais de entrada de interrupção do microprocessador, os dispositivos periféricos podem informar a ocorrência de um

dado evento externo, que será tratado com maior facilidade e rapidez através das **subrotinas de tratamento de interrupção**, sem muito comprometer o tempo de execução do programa que estava em execução. Ou seja, as condições internas do microprocessador são preservadas para posterior continuação da execução do programa interrompido (*figura 1*). Com esse método, o microprocessador não necessita consumir tempo de processamento para pesquisar a ocorrência de eventos externos, e pode garantir um tempo reduzido para iniciar o seu tratamento.

Exemplos: sistemas com muitos dispositivos periféricos, uso de dispositivos cujos eventos envolvidos são sempre sinalizados assincronamente (canais de comunicação serial ou paralelos, temporizadores e contadores programáveis, controladores de periféricos, relógios de tempo real, sinais de emergência, sinais de alguns sensores pulsados, etc).
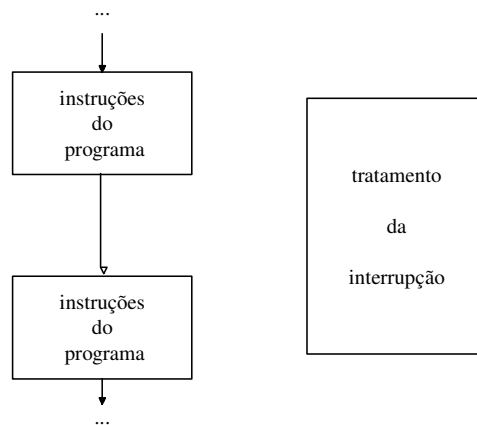
Figura 1 – Técnica de Transferência de Dados de Entrada e Saída via Interrupção

3. MECANISMOS DE INTERRUPÇÃO

3.1. Características Básicas de Interrupções

Como foi apresentado, as interrupções possibilitam um tratamento mais eficiente das operações de entrada e saída, e permitem melhor atender aos requisitos de aplicações em tempo real.

A requisição de interrupção pode ocorrer a qualquer momento (assincronamente), sendo ela indicada pela ativação de um "*flag*" pelo dispositivo periférico. O microprocessador reconhece a interrupção, enviando sinais de controle, completa a execução da instrução corrente, salva o conteúdo dos registradores de interesse (contador de programa, status, etc), e atende ao dispositivo periférico que solicitou a interrupção, transferindo o controle para a subrotina de tratamento da interrupção. Ao término da execução desta, o microprocessador desativa o "*flag*" de indicação de interrupção, restaura os registradores que foram salvos, e transfere o controle para a instrução seguinte ao ponto de interrupção do programa.

Algumas interrupções podem ter seu tratamento postergado, enquanto que outras necessitam de tratamento imediato (por exemplo, sinais de emergência). Em função disso, a maioria do microprocessadores apresentam interrupções **mascaráveis** e interrupções **não-mascaráveis**. Através de instruções apropriadas o programa pode habilitar ou desabilitar uma interrupção mascarável, enquanto que a não-mascarável deverá ser sempre atendida, devendo ser reservada apenas para eventos de alta importância. Assim, aplicações de tempo real que envolvem módulos de programas críticos, e que não podem ser interrompidos durante sua execução, devem utilizar o conceito de mascaramento.

Outra característica das interrupções diz respeito à forma de obtenção dos endereços das subrotinas de tratamento, que varia de um microprocessador para outro. Ela costuma ser do tipo ***fixa***, ***vetorada*** ou ***não-vetorada***.

- **interrupções fixas:**

  Requerem um "hardware" relativamente simples de interface. Um "*flag"* é ativado pela linha de interrupção, indicando a requisição. Caso apenas um dispositivo periférico esteja ligado a essa linha, então o controle é transferido para um posição fixa de memória, que apresenta a subrotina de tratamento da interrupção. Quando vários dispositivos são acoplado a uma mesma linha, cada um com sua própria subrotina de tratamento, o microprocessador tem que identificar o dispositivo periférico responsável pela interrupção. Para contornar essa dificuldade, pode-se utilizar múltiplas linhas, uma para cada interrupção, tendo-se uma posição fixa de memória para a colocação da subrotina de tratamento de cada uma. Entretanto, existe a necessidade de tantas linhas quantos forem os dispositivos periféricos que solicitam interrupção. Exemplos: interrupções RST5.5, RST6.5, RST7.5 e TRAP do 8085; interrupção NMI/ do Z80.

- **interrupções vetoradas:**

  Este tipo exige a identificação do dispositivo periférico que solicita interrupção. Essa identificação é utilizada para a localização do endereço da subrotina de tratamento da interrupção, em uma tabela localizada em uma região determinada da memória do **microprocessador (vetor de interrupções**). Exemplo: interrupções da família 68000.

- **interrupções não-vetoradas:**

  Nesse tipo, o dispositivo periférico fornece ao microprocessador diretamente o endereço de subrotina de tratamento da interrupção (por exemplo, as interrupções do 8086/8088). Em alguns sistemas, o dispositivo periférico fornece apenas metade de endereço, estando a outra metade armazenada em um registrador do microprocessador, carregado na fase de inicialização do sistema (por exemplo, as interrupções do Z80 operando no modo 2). Em outros sistemas, o dispositivo periférico tem que fornecer uma instrução ao microprocessador, normalmente a instrução de chamada da subrotina de tratamento da interrupção (por exemplo, a interrupção INTR do 8085, e INT/ do Z80 operando no modo 0).

Em sistemas com mais de uma interrupção, existe a possibilidade de ocorrerem pedidos de interrupção simultâneos, devendo existir algum critério para a escolha de qual será atendida em primeiro lugar. Esse critério, normalmente por prioridade, pode ser implementado de diversas formas: "*daisy chain*" ou circuitos de prioridade.

- **"*daisy chain*":**

  Nesse esquema, os dispositivos periféricos que podem solicitar interrupção são interligados, através de sinais de controle, em uma cadeia conhecida como "*daisy chain*", sendo o primeiro elemento ligado ao microprocessador. Em caso de pedido de interrupção, o microprocessador envia um sinal ao primeiro elemento; caso seja ele o autor do pedido, ele responde com o endereço da subrotina de tratamento; caso contrário, ele repassa o pedido ao próximo dispositivo periférico, que repetirá o procedimento descrito. Dessa maneira, o primeiro elemento da cadeia deverá corresponder ao de maior prioridade, e assim sucessivamente. Esse esquema pode ser implementado, por exemplo, com o Z80 operando no modo 2.

- **circuitos de prioridade:**

  Nesse esquema, um codificador de prioridades é utilizado (por exemplo, de 8 para 3), fornecendo em sua saída o código do pedido da interrupção de maior prioridade presente na entrada. Esse código pode ser inspecionado pelo microprocessador para descobrir qual interrupção atender. Muitas vezes são disponíveis esquemas para o mascaramento individual das interrupções, permitindo que uma interrupção alocada em uma linha de maior prioridade não seja atendida quando outra de menor prioridade ocorrer. Esse esquema pode ser implementado, por exemplo, com 8085, Z80 operando no modo 0, e a família MC68000.

3.2. Interrupções na Família 8051

Cada família de microprocessadores apresenta um tipo de implementação para as suas interrupções, sendo encontradas muitas particularidades não necessariamente encontradas em outras famílias. Recomenda-se, nesse ponto, que seja efetuada uma leitura atenta nas seguintes referências:

- Capítulo 2 de [1].
- Nota AN420 de [2].
- Capítulo 6 de [8].
- Capítulo 6 de [9].
- Páginas 10 e 11 de [7] no tocante aos nomes de "bits" e "registradores" para configurações em geral.

A seguir, serão resumidos os principais conceitos envolvidos.

O 8051 dispõe de cinco fontes de interrupção: duas interrupções externas (INT0 e INT1), duas interrupções por "timer" (Timer0 e Timer1) e uma interrupção do "port" serial.

Cada fonte de interrupção pode ser habilitada ou desabilitada individualmente, através da colocação de valor lógico "um" ou "zero" em um determinado "bit" do registrador **IE ("Interrupt Enable")**. Este registrador também tem um "bit" de habilitação global, que permite que todas de uma vez sejam habilitadas. Este "bit" deve estar corretamente inicializado para que então se possam realizar habilitações individuais para cada tipo de interrupção. Além disto cada fonte pode ter um ou dois níveis de prioridade, de acordo com uma inicialização colocada em um "bit" do Registrador **IP ("Interrupt Priority")** [1].

As interrupções externas INT0 e INT1 podem ser programadas como sensíveis a nível ou sensíveis a borda, dependendo da programação dos "bits" **IT0** e **IT1** no registrador **TCON**. Os "flags" gerados com estas interrupções estão disponíveis nos "bits" **IE0** e **IE1** no mesmo registrador **TCON**.

A tabela com o endereço onde devem estar as rotinas de tratamento das interrupções encontra-se em [4].

4. BIBLIOGRAFIA

[1] PHILIPS; 80C51-Based 8-Bit Microcontrollers – Data Handbook IC20, Philips Electronics North America Corporation, USA, 1.997.

[2] PHILIPS; Application Notes and Development Tools for 80C51 – Data Handbook, Philips Electronics North America Corporation, USA, 1.997.

[3] Gomi, Edson Satoshi; Apostila da Experiência Microprocessadores, Disciplina PCS 308 - Laboratório Digital II, 1.998.

[4] Hirakawa, André; Cugnasca, Carlos Eduardo; Apostila da Experiência Familiarização com a Placa Experimental de Microcontrolador, Disciplina PCS 598 - Laboratório de Microprocessadores II, 1.999.

[5] Cugnasca, Carlos Eduardo; Hirakawa, André; Apostila da Experiência Interface com Teclado e Display, Disciplina PCS 598 - Laboratório de Microprocessadores II, 1.999.

[6] Andréa M. Matsunaga; Maurício 0. Tsugawa; Projeto de Formatura Sistema de Pesagem Dinâmica. PCS-588 – Laboratório de Projeto de Formatura, 1.997.

[7] 2500AD Software 8044/51; X8051 Cross Assembler User Manual.

[8] Silva Júnior, Engenheiro Vidal Pereira da; Aplicações Práticas do Microcontrolador 8051 – Hardware & Software, Editora Érica Ltda., 1.990.

[9] Silva Júnior, Engenheiro Vidal Pereira da; Aplicações Práticas do Microcontrolador 8051 – Teoria Geral, Editora Érica Ltda., 1.994.

[10] ALFACOM; Módulos Multi-Matrix – Manual de Utilização, Editora Érica Ltda., 1.994.

[11] MC8051 – Diagrama Lógico da Placa Experimental do 8051

5. PARTE EXPERIMENTAL

5.1 Implementação de um Cronômetro Digital

Antes de iniciar o planejamento desta experiência, é fundamental a familiarização com os mecanismos de interrupção e a programação do timer do microprocessador 8051 [1][2].

Com os conhecimentos e recursos de utilização do teclado vistos nas experiências 2 ("Display de Cristal Líquido") e 3 ("Interface com Teclado e Display") e os resultados do item 5.1, pede-se que seja implementado um cronômetro digital que faça a contagem de 10 segundos, com o seguinte comportamento:

- O início da contagem deve ser dado pelo pressionar de uma tecla no Kit, quando deverá ser mostrada no display a mensagem "**S**" (de *Start*) na posição 0 da primeira linha do display.
- A cada 1 segundo o display deverá ser atualizado, sempre na mesma posição, com os valores **1**, **2**, **3**, **4**, **5**, **6**, **7**, **8** e **9**.
- Ao final dos 10 segundos, a mensagem **E** (de *End*) deverá ser apresentada.

Pede-se que se implemente um Cronômetro Digital, fazendo-se uso de interrupção do Timer 0 **TF0** (modo 2) para gerar a base de tempo. Siga as seguintes etapas para o projeto, implementação e teste do programa.

a) Utilize as subrotinas básicas de interfaceamento com o Display e o Teclado desenvolvidas nas experiência 2 e 3.

b) Usar como fonte de referência de tempo um **centésimo de segundo**, obtido pela contagem das chegadas dos sinais de interrupção, proveniente do Timer 0.

c) Quando do desenvolvimento da subrotina para tratamento de interrupção, deve-se lembrar de que há um endereço correto onde esta rotina deve estar carregada. Descreva a rotina e indique o endereço de carga.

d) A subrotina para tratamento de interrupção deve se limitar a incrementar um registrador que conta eventos e, eventualmente, incrementa um contador de centésimos de segundos. A atualização do display deve ser realizada no programa principal, a cada 1 segundo.

5.2 Observações

O planejamento deverá apresentar:

- Descrição do projeto, relacionando suas características principais.
- Especificação de cada uma das subrotinas através de **diagramas estruturados.**

Os códigos dos programas, com comentários, **podem ser elaborados no laboratório**.

***Sugestão**: Utilizando o modo 2 do Timer 0 (auto-load), iniciando seus registradores com 00h, as interrupções ocorrerão a uma taxa de 1/3600 segundo. Logo, a cada 36 interrupções terá se passado 1 centésimo de segundo. Faça o cálculo de forma explícita no relatório, demonstrando o cálculo.*